**Содержание**

[ВВЕДЕНИЕ 10](#_Toc168015623)

[ГЛАВА 1 Аналитическая часть 12](#_Toc168015624)

[1.1 Обзор и анализ 12](#_Toc168015625)

[1.2 Анализ требований 14](#_Toc168015626)

[1.3 Анализ спецификации ИС. Постановка задачи 17](#_Toc168015627)

[1.4 Анализ и обоснование выбора технологии проектирования 23](#_Toc168015628)

[1.5 Обоснование среды разработки 35](#_Toc168015629)

[Глава 2 Проектная часть 37](#_Toc168015630)

[2.1 Концептуальная модель информационных потоков данных 37](#_Toc168015631)

[2.2 Функциональная модель информационных потоков данных 39](#_Toc168015632)

[2.3 Логическая схема взаимодействия отдельных частей БД 40](#_Toc168015633)

[2.4 Физическая схема взаимодействия отдельных частей БД 41](#_Toc168015634)

[2.5 Функциональная модель 43](#_Toc168015635)

[2.6 Схема технологического процесса 43](#_Toc168015636)

[2.7 Схема взаимосвязи программных модулей 45](#_Toc168015637)

[2.6 Подробное описание проектирования и разработки приложения 47](#_Toc168015638)

[Глава 3 Разработка документации 49](#_Toc168015639)

[3.1 Руководство пользователю 49](#_Toc168015640)

[3.2 Руководство программисту 54](#_Toc168015641)

[3.3. Тестирование системы 57](#_Toc168015642)

[ГЛАВА 4 Технико-экономическое обоснование разработки программного продукта, который должен содержать следующие подразделы 59](#_Toc168015643)

[4.1 Определение суммарных затрат на разработку 59](#_Toc168015644)

[4.2 Расчет ориентировочной цены 67](#_Toc168015645)

[ЗАКЛЮЧЕНИЕ 68](#_Toc168015646)

[СПИСОК ЛИТЕРАТУРЫ 69](#_Toc168015647)

# ВВЕДЕНИЕ

В современной сфере бизнеса, где конкуренция и эффективность играют ключевую роль, создание и внедрение информационной системы управления отношениями с клиентами (CRM) для строительных компаний, принимающих заявки на аренду строительной техники, становится стратегически важным шагом. CRM-система обеспечивает автоматизацию и упрощение процессов взаимодействия с клиентами, включая управление контактами, обработку запросов, анализ поведения клиентов и обеспечение качественного обслуживания.

Целью разработки и внедрения CRM-системы для строительных компаний является улучшение взаимодействия с клиентами, повышение их удовлетворенности и увеличение объема аренды строительной техники. Это достигается за счет автоматизации рутинных процессов, предоставления персонализированных услуг и анализа данных для принятия обоснованных решений.

Задачи CRM-системы включают управление контактами, обработку запросов, анализ поведения клиентов, обслуживание клиентов, маркетинг и продажи. Для их решения используются методы автоматизации, предоставления персонализированных услуг и анализа данных.

Разработка CRM-системы для строительных компаний также включает в себя интеграцию с другими информационными системами, создание мобильных и веб-приложений, обеспечение безопасности данных, а также аналитику, отчетность, обучение и поддержку пользователей, а также тестирование и внедрение системы.

Внедрение CRM-системы для строительных компаний требует комплексного подхода, включающего планирование, разработку, тестирование и поддержку. Учитывая современные технологии и требования рынка, CRM-система может стать ключевым элементом успеха компании, улучшая отношения с клиентами и повышая эффективность бизнес-процессов.

Выбор технологий для разработки CRM-системы зависит от специфики бизнеса, требований к функциональности, бюджета и доступных ресурсов. В настоящее время широко используются технологии, такие как JavaScript (JS) и React для разработки пользовательского интерфейса, а также SQL и Node.js для работы с базами данных и серверной логикой, обеспечивая высокую производительность, гибкость и масштабируемость системы.

Объектом исследования в данном случае является сама CRM-система, используемая для управления отношениями с клиентами в строительных компаниях. Предметом исследования является комплекс программных и аппаратных средств, обеспечивающих эффективное взаимодействие и обмен информацией между компанией и ее клиентами.

Целью проекта является создание и внедрение CRM-системы, способствующей улучшению отношений с клиентами, повышению уровня их удовлетворенности и увеличению объема аренды строительной техники. Основная задача заключается в автоматизации и оптимизации процессов взаимодействия с клиентами, что позволит строительной компании оставаться конкурентоспособной на рынке.

# ГЛАВА 1 Аналитическая часть

## 1.1 Обзор и анализ

Обзор и анализ существующих информационных систем для приема заявок на аренду строительной техники показывает, что выбор подходящей системы зависит от нескольких ключевых факторов. Эти факторы включают в себя функциональность, масштабируемость, интеграцию с уже используемыми системами, а также вопросы безопасности и поддержки со стороны разработчика. Ниже представлен обзор нескольких популярных информационных систем:

1. SAP ERP

SAP ERP предлагает комплексное решение для управления бизнес-процессами, включая учет и финансы, закупки, производство и распределение. Эта система обеспечивает высокую степень интеграции между различными функциональными областями и масштабируемость под потребности компании. Однако, стоимость внедрения и поддержки может быть значительной.

2. Microsoft Dynamics 365 for Finance and Operations

Microsoft Dynamics 365 предлагает модуль для управления финансами и операциями, который включает функционал управления финансами, управления запасами, производства и дистрибуции. Это облачное решение обеспечивает гибкость и масштабируемость, а также интеграцию с другими приложениями Microsoft. Однако, как и в случае с SAP, стоимость может быть значительной для малых и средних предприятий.

3. Odoo

Odoo предлагает открытую платформу для управления бизнес-процессами, которая включает в себя модули для учета, продаж, закупок, склада и производства. Эта система отличается гибкостью и возможностью настройки под конкретные потребности компании. Однако, интеграция с другими системами может потребовать дополнительных усилий.

Выбор технологий для разработки информационной системы:

Для разработки пользовательского интерфейса и клиентской части информационной системы рекомендуется использовать современные технологии, такие как JavaScript (JS) и React. Эти инструменты обеспечивают высокую производительность, модульность и повторное использование кода, что упрощает разработку и поддержку системы.

Для серверной разработки и управления данными рекомендуется использовать технологии, такие как SQL и Node.js. SQL обеспечивает надежное управление данными в базах данных, а Node.js позволяет разрабатывать масштабируемые и быстрые веб-приложения.

Выбор подходящей информационной системы для приема заявок на аренду строительной техники требует тщательного анализа и учета специфики бизнеса, требований к функциональности и бюджета. При этом использование современных технологий позволит создать эффективное и масштабируемое решение, способное удовлетворить потребности компании и её клиентов.

Дополнительные аспекты разработки информационной системы для приема заявок на аренду строительной техники включают в себя:

1. Интеграция с системами управления складом и учетом. Это позволит автоматизировать процессы отслеживания запасов строительной техники, учета аренды и финансовых операций, обеспечивая единый и целостный подход к управлению ресурсами компании.

2. Автоматизация процессов обработки заявок. Внедрение функционала автоматической обработки заявок позволит ускорить процесс принятия решений и выполнения заказов, уменьшить вероятность ошибок и повысить уровень обслуживания клиентов.

3. Интеграция с онлайн-платежами. Обеспечение возможности онлайн-оплаты аренды строительной техники упростит процесс взаиморасчетов между компанией и клиентами, повысит удобство использования услуг и сократит время на заключение сделок.

4. Реализация системы отчетности и аналитики. Внедрение функционала аналитики позволит компании анализировать данные о заказах, доходах, клиентах и использовании техники, что поможет оптимизировать бизнес-процессы, выявить тренды и принимать обоснованные решения для развития бизнеса.

5. Обеспечение безопасности данных. Защита конфиденциальности и целостности данных клиентов и компании является критически важным аспектом разработки информационной системы. Необходимо применять современные методы шифрования, установить системы мониторинга безопасности и обеспечить соблюдение всех требований к защите данных.

6. Проведение обучения и поддержки пользователей. После внедрения системы важно обеспечить обучение сотрудников по её использованию и обеспечить непрерывную техническую поддержку для решения возникающих проблем и вопросов.

7. Тестирование и внедрение. Перед внедрением системы в эксплуатацию необходимо провести тщательное тестирование всех её компонентов, а также проверить её работоспособность в реальных условиях. Планирование и контроль процесса внедрения также играют важную роль в успешном запуске новой информационной системы.

## 1.2 Анализ требований

Анализ требований к созданию информационной системы для строительных компаний, предназначенной для приема заявок на аренду строительной техники, направлен на выявление ключевых функциональных и технических характеристик, необходимых для эффективного управления процессом приема и обработки заявок от клиентов. Ниже представлен подробный анализ требований к информационной системе:

1. Управление клиентскими контактами и заявками:

Реализация возможности создания, редактирования и удаления контактной информации клиентов.

Система должна обеспечить механизм управления заявками, начиная с их поступления и отслеживания статуса до завершения сделки.

Важна возможность классификации и категоризации клиентов для эффективного поиска и организации информации.

2. Управление заказами и обслуживание клиентов:

Система должна поддерживать процесс отслеживания заказов, включая создание, управление и отслеживание статуса выполнения заказов, а также обработку возвратов.

Реализация системы обслуживания клиентов для эффективной обработки запросов, жалоб клиентов и предоставления поддержки через различные каналы связи.

3. Маркетинговые возможности и аналитика:

Система должна предоставлять инструменты для проведения маркетинговых кампаний, включая управление кампаниями, отслеживание результатов и анализ поведения клиентов.

Необходимо обеспечить возможность анализа данных о заказах и взаимодействии с клиентами для выявления трендов и оптимизации маркетинговых стратегий.

4. Обеспечение безопасности и конфиденциальности данных:

Система должна гарантировать безопасность и конфиденциальность данных клиентов, включая аутентификацию пользователей, контроль доступа и шифрование информации.

Необходимы меры защиты от несанкционированного доступа и утечки данных.

5. Интеграция с другими системами:

Информационная система должна быть способной интегрироваться с другими системами, такими как системы управления контентом (CMS), системы электронной коммерции (e-commerce) и системы управления проектами (PM).

Важна возможность обмена данными между различными системами для синхронизации информации и улучшения взаимодействия между различными отделами компании.

6. Отчетность и аналитика:

Система должна предоставлять возможность генерации отчетов о заказах, клиентском поведении и эффективности маркетинговых кампаний.

Важна возможность анализа данных для выявления трендов и оптимизации бизнес-процессов.

7. Пользовательский интерфейс:

Интерфейс информационной системы должен быть интуитивно понятным и удобным в использовании как для сотрудников компании, так и для клиентов.

Важно обеспечить адаптивность системы для работы на различных устройствах и платформах.

8. Масштабируемость и гибкость:

Информационная система должна быть масштабируемой и гибкой, способной адаптироваться к изменяющимся потребностям компании и росту объема данных.

Важна возможность добавления новых функций и интеграции с другими системами без значительных изменений в архитектуре.

9. Поддержка и обучение:

После внедрения системы необходимо предоставить обучение сотрудникам для эффективного использования системы.

Важна поддержка пользователей и техническая поддержка для решения возникающих вопросов и проблем.

Анализ требований к информационной системе для приема заявок на аренду строительной техники позволяет определить ключевые функции и характеристики, необходимые для создания эффективной системы управления заявками и обслуживания клиентов. Важно учитывать, что конкретные требования могут различаться в зависимости от особенностей компании и потребностей её клиентов. При разработке информационной системы рекомендуется провести более детальный анализ требований, учитывая специфику бизнеса и его целей.

## 1.3 Анализ спецификации ИС. Постановка задачи

### 1.3.1 Цель и назначение создания и модернизации модулей или сервисов информационной системы. Функциональные требования

Цель и назначение создания и модернизации информационной системы для приема заявок на аренду строительной техники для строительных компаний

Цель создания и модернизации

Главная цель создания и модернизации информационной системы для приема заявок на аренду строительной техники заключается в оптимизации процессов управления, повышении удобства использования и обеспечении безопасности платформы. Важные аспекты этой цели включают:

1. Улучшение опыта пользователей:

Оптимизация интерфейса и путей навигации для удобства клиентов.

Обеспечение быстрого и точного поиска нужной строительной техники.

Предоставление персонализированных рекомендаций в соответствии с потребностями клиентов.

2. Повышение эффективности управления:

Автоматизация бизнес-процессов, включая управление запасами, обработку заказов и логистику.

Улучшение аналитики и отчетности для принятия обоснованных решений.

Интеграция с внешними системами, такими как системы платежей, доставки, CRM и ERP.

3. Обеспечение безопасности:

Защита данных пользователей и финансовых операций.

Обновление систем безопасности в соответствии с современными стандартами и требованиями законодательства.

4. Расширение функциональности:

Внедрение новых функций, таких как мобильные приложения, интеграция с социальными сетями и дополнительные способы оплаты.

Обеспечение масштабируемости и готовности к расширению бизнеса.

Назначение модулей и сервисов

Назначение модулей и сервисов информационной системы для приема заявок на аренду строительной техники включает следующие аспекты:

1. Модуль управления инвентарем:

Добавление, редактирование и удаление техники.

Управление категориями, характеристиками и доступностью техники.

Интеграция с системой учета склада для автоматического обновления доступности.

2. Модуль заказов и оформления:

Возможность добавления техники в корзину и оформления заказа.

Выбор методов оплаты и доставки.

Автоматический расчет стоимости доставки и налогов.

3. Модуль платежей:

Интеграция с различными платежными системами.

Обеспечение безопасности финансовых операций с использованием современных технологий шифрования.

Управление возвратами и платежами.

4. Модуль управления клиентами:

Регистрация и аутентификация клиентов.

Управление учетными записями и профилями.

Персонализация интерфейса и предложений.

5. Модуль аналитики и отчетности:

Сбор и анализ данных о заказах, посещениях и поведении пользователей.

Генерация отчетов для принятия решений.

Визуализация ключевых показателей производительности.

6. Модуль маркетинга:

Управление акциями и скидками.

Интеграция с email-маркетингом и социальными сетями.

Внедрение программ лояльности и реферальных систем.

7. Модуль поддержки клиентов:

Предоставление каналов связи для клиентов.

Управление запросами на возврат и обмен техники.

Интеграция с CRM для отслеживания взаимодействия с клиентами.

Таким образом, создание и модернизация информационной системы направлены на обеспечение удобства, безопасности и эффективности для всех участников процесса аренды строительной техники.

Кроме того, функциональные требования к информационной системе для приема заявок на аренду строительной техники также включают в себя:

8. Интеграция с системами управления техникой:

Обеспечение возможности интеграции с системами управления и мониторинга техники, чтобы клиенты могли отслеживать состояние и местоположение арендованной техники в реальном времени

9. Модуль управления расписанием и доступностью:

Реализация функционала для управления расписанием аренды техники и ее доступностью, что позволит клиентам бронировать необходимое оборудование заранее и избежать конфликтов с другими заказами.

10. Модуль обработки документов:

Имплементация системы для загрузки и обработки необходимых документов, таких как договоры аренды и сопроводительные документы, чтобы сделать процесс аренды более прозрачным и безбумажным.

11. Модуль обучения и поддержки:

Предоставление клиентам обучающих материалов и ресурсов для эффективного использования арендованной техники, а также технической поддержки в случае возникновения проблем или вопросов.

12. Модуль управления инцидентами и авариями:

Разработка функционала для регистрации и отслеживания инцидентов и аварий, связанных с арендованной техникой, с возможностью оперативного реагирования и предоставления помощи.

Все эти модули и сервисы предназначены для создания полноценной и интегрированной информационной системы, которая обеспечит строительным компаниям простой и эффективный процесс аренды строительной техники, а также повысит удовлетворенность клиентов и уровень сервиса.

Функциональные требования к информационной системе для строительных компаний, предназначенной для приема заявок на аренду строительной техники, определяют ключевые функции и характеристики, необходимые для создания эффективной платформы управления процессом аренды. Важно отметить, что данные требования могут варьироваться в зависимости от особенностей компании и потребностей её клиентов. При разработке данной информационной системы рекомендуется провести более детальный анализ требований, учитывая специфику организации и её цели.

1.Модуль приема заявок и заказов: Этот модуль должен обеспечивать простой и удобный способ для клиентов строительной компании отправлять заявки на аренду строительной техники. Он должен предоставлять возможность заполнения формы с указанием необходимой техники, даты и времени аренды, а также контактной информации заказчика. Помимо этого, система должна автоматически генерировать уведомления о новых заявках для оперативного реагирования со стороны сотрудников компании.

2.Управление инвентарем и доступностью техники: Для эффективной работы системы необходим модуль, позволяющий отслеживать доступность строительной техники для аренды. Этот модуль должен вести учет имеющегося инвентаря, его технического состояния и графика занятости. Информация о доступности должна быть доступна как для клиентов при оформлении заявок, так и для сотрудников компании для оперативного управления заказами.

3.Отчетность и аналитика: Для повышения эффективности работы и принятия обоснованных бизнес-решений, важно включить в систему модуль для формирования отчетов и аналитики. Этот модуль должен предоставлять возможность генерации отчетов о количестве заявок, выручке, наиболее востребованных видов техники и других ключевых метриках. Аналитические инструменты позволят строительной компании оценить эффективность своей работы и выявить потенциальные улучшения в процессе аренды строительной техники.

### 1.3.2. Анализ требований к информационно-коммуникационным средствам для решения задачи. Нефункциональные требования.

Нефункциональные требования играют важную роль в разработке информационной системы для строительных компаний, предназначенной для приема заявок на аренду строительной техники. Эти требования определяют характеристики системы, влияющие на ее производительность, надежность, доступность и безопасность. Давайте рассмотрим основные нефункциональные требования к этой информационной системе:

1. Производительность и масштабируемость:

Производительность: Система должна обеспечивать быструю загрузку страниц и минимальные задержки при обработке заявок на аренду. Время загрузки страницы не должно превышать 2 секунды.

Масштабируемость: Система должна быть способной масштабироваться как горизонтально, так и вертикально, чтобы обеспечивать стабильную работу при увеличении числа пользователей и объема данных.

2. Надежность и доступность:

Надежность: Система должна быть стойкой к сбоям и иметь механизмы автоматического восстановления после отказов. Резервное копирование данных должно выполняться регулярно.

Доступность: Система должна быть доступной для пользователей круглосуточно с уровнем доступности не менее 99,9%. Плановые технические работы должны проводиться во время минимальной активности пользователей.

3. Безопасность:

Конфиденциальность данных: Личные и финансовые данные пользователей должны быть защищены от несанкционированного доступа с использованием современных методов шифрования и аутентификации.

Защита от атак: Система должна быть защищена от различных видов кибератак, таких как DDoS-атаки, SQL-инъекции и XSS. Регулярное обновление программного обеспечения и проведение аудитов безопасности необходимы.

4. Удобство использования и эргономичность:

Интерфейс пользователя: Интерфейс должен быть интуитивно понятным и удобным для пользователей, соответствуя современным стандартам веб-дизайна.

Многоязычность и локализация: Система должна поддерживать несколько языков и легко адаптироваться к разным регионам.

5. Совместимость и интеграция:

Кроссбраузерность: Система должна корректно работать во всех популярных веб-браузерах и на различных устройствах.

Интеграция с внешними системами: Система должна поддерживать интеграцию с различными внешними сервисами, такими как платежные шлюзы и CRM.

6. Поддержка и обслуживание:

Документированность: Система должна иметь полную документацию, включая техническую информацию для разработчиков и инструкции для пользователей.

Техническая поддержка: Наличие службы поддержки для оперативного решения проблем пользователей.

7. Мобильность:

Мобильная оптимизация: Система должна быть оптимизирована для мобильных устройств, обеспечивая удобство использования на смартфонах и планшетах.

Мобильный адаптив: Поддержка создания мобильных приложений для iOS и Android с дополнительным функционалом для пользователей.

8. Экономическая эффективность:

Затраты на обслуживание: Минимизация затрат на обслуживание системы за счет использования современных технологий.

Возврат инвестиций: Обеспечение высокой эффективности работы системы для повышения возврата инвестиций.

Эти нефункциональные требования гарантируют высокое качество информационной системы для строительных компаний, делая ее надежной, безопасной и удобной в использовании.

## 1.4 Анализ и обоснование выбора технологии проектирования

### 1.4.1 Язык программирования JavaScript

JavaScript (JS) - это высокоуровневый, интерпретируемый язык программирования, который широко используется для разработки веб-приложений и динамических веб-страниц. Он был создан для придания интерактивности и динамического поведения веб-страницам. Вот подробное описание основных характеристик и возможностей JavaScript:

1. Синтаксис и типы данных:

- JavaScript[[5]](#_СПИСОК_ЛИТЕРАТУРЫ) имеет синтаксис, похожий на другие языки программирования, такие как C, C++ и Java. Это делает его относительно простым для изучения и использования.

- В языке поддерживаются различные типы данных, включая числа, строки, булевы значения, массивы, объекты, null и undefined. Кроме того, JavaScript предоставляет возможности работы с регулярными выражениями и датами.

2. Объектно-ориентированное программирование (ООП):

- JavaScript является прототипно-ориентированным языком, где объекты создаются с использованием функций-конструкторов или литералов объектов.

- Язык поддерживает наследование, инкапсуляцию и полиморфизм, что является ключевыми принципами объектно-ориентированного программирования.

- С помощью прототипов JavaScript позволяет создавать цепочки наследования и повторно использовать код.

3. Функции:

- Функции в JavaScript являются объектами первого класса. Они могут быть назначены переменным, переданы как аргументы и возвращены из других функций.

- JavaScript поддерживает как именованные функции, так и анонимные функции, а также возможность определения функций внутри других функций (замыкания).

- Функции в JavaScript могут быть вызваны с помощью различных методов, включая вызов в качестве конструктора с использованием оператора "new".

4. Управление потоком выполнения:

- JavaScript предлагает различные конструкции для управления потоком выполнения кода, включая условные операторы (if-else, switch) и циклы (for, while, do-while).

- Он также поддерживает обработку исключений с использованием оператора try-catch, что позволяет перехватывать и обрабатывать ошибки и исключения.

5. Взаимодействие с HTML и CSS:

- JavaScript позволяет взаимодействовать с элементами HTML и стилями CSS на веб-странице, изменять их содержимое, атрибуты и стили.

- С помощью JavaScript можно обрабатывать события, добавлять и удалять элементы, изменять классы и анимировать элементы, что делает веб-страницы более интерактивными и динамичными.

6. Библиотеки и фреймворки:

- JavaScript имеет богатую экосистему библиотек и фреймворков, которые облегчают разработку и расширяют возможности языка.

- Некоторые популярные библиотеки и фреймворки включают React, Angular, Vue.js, Node.js, Express.js и многие другие.

- Эти инструменты предоставляют готовые решения для разработки веб-приложений, управления состоянием, маршрутизации и других задач.

JavaScript является мощным языком программирования, который предлагает широкий набор возможностей для разработки веб-приложений. Его гибкость, простота использования и широкая поддержка делают его одним из наиболее популярных языков программирования в мире. JavaScript применяется в различных сферах, включая веб-разработку, серверное программирование, разработку мобильных приложений и интернета вещей, что подчеркивает его важность и актуальность в современном мире программирования.

Консоль разработчика: Консоль разработчика веб-браузера (например, Chrome DevTools) является незаменимым инструментом для отладки JavaScript-кода. Она позволяет проверять значения переменных, исправлять ошибки, анализировать производительность и многое другое.

Сборщики модулей: Сборщики модулей, такие как webpack и Rollup, позволяют разбивать JavaScript-код на модули и объединять их в один оптимизированный файл. Это упрощает организацию и структурирование кода, а также улучшает производительность и скорость загрузки приложения.

Фреймворки и библиотеки: Существуют множество фреймворков и библиотек JavaScript, которые облегчают разработку и предоставляют готовые решения для различных задач. Некоторые из популярных фреймворков включают React.js, Angular.js, Vue.js, Express.js (для серверной разработки) и Node.js.

ECMAScript: ECMAScript (ES) - это стандарт, на основе которого разрабатывается JavaScript. Новые версии ECMAScript, такие как ES6, ES7, ES8 и т.д.вошли значительные улучшения в язык, включая новые возможности, синтаксический сахар, улучшенную поддержку модулей и другие функции. Эти обновления позволяют разработчикам писать более чистый и эффективный код.

WebAssembly: WebAssembly (Wasm) - это двоичный формат, который позволяет запускать высокопроизводительный код на веб-платформе. JavaScript может взаимодействовать с WebAssembly-модулями, предоставляя возможность создавать более мощные и быстрые веб-приложения.

Progressive Web Apps: Progressive Web Apps (PWA) объединяют возможности веб-сайтов и мобильных приложений. JavaScript играет важную роль в разработке PWA, позволяя создавать оффлайн-функциональность, отправлять уведомления и обеспечивать более плавный и реактивный пользовательский интерфейс.

### 1.4.2. Cтандартный язык разметки HTML

HTML[[6]](#_СПИСОК_ЛИТЕРАТУРЫ) (HyperText Markup Language) является стандартным языком разметки для создания веб-страниц. Он используется для описания структуры и содержания веб-документов. HTML-документ состоит из различных элементов, которые определяют типы содержимого и их отображение в браузере.

Основные компоненты HTML:

1. Элементы:

Элементы HTML представляют собой строительные блоки, используемые для разметки различных частей веб-страницы. Каждый элемент обозначается с помощью тега, который состоит из открывающего и закрывающего тегов. Между этими тегами находится содержимое элемента.

Пример элемента:

```html

<p>Это абзац текста.</p>

2. Атрибуты:

Атрибуты HTML добавляют дополнительные сведения к элементам и задают им различные свойства. Атрибуты указываются в открывающем теге элемента и содержат имя и значение, разделенные знаком равенства.

Пример использования атрибута:

```html

<a href="https://www.example.com">Это ссылка на примерный сайт</a>

3. Текстовое содержимое:

HTML позволяет размечать текстовое содержимое веб-страницы. Вы можете использовать различные теги для выделения текста, создания заголовков, создания списков и т.д.

Примеры текстового содержимого:

```html

<h1>Заголовок первого уровня</h1>

<p>Это абзац текста.</p>

<ul>

<li>Элемент списка 1</li>

<li>Элемент списка 2</li>

</ul>

4. Изображения:

HTML позволяет вставлять изображения на веб-страницу с помощью тега `<img>`. Вы можете указать источник изображения, его размеры и альтернативный текст.

Пример вставки изображения:

```html

<img src="image.jpg" alt="Описание изображения" width="300" height="200">

5. Ссылки:

HTML предоставляет тег `<a>`, который используется для создания ссылок. Вы можете указать URL-адрес ссылки и текст, который будет отображаться как ссылка.

Пример создания ссылки:

```html

<a href="https://www.example.com">Это ссылка на примерный сайт</a>

6. Таблицы:

HTML позволяет создавать таблицы для представления структурированных данных. Вы можете использовать теги `<table>`, `<tr>` и `<td>` для определения структуры таблицы и размещения данных в ячейках.

Пример создания таблицы:

```html

<table>

<tr>

<td>Ячейка 1</td>

<td>Ячейка 2</td>

</tr>

<tr>

<td>Ячейка 3</td>

<td>Ячейка 4</td>

</tr>

</table>

Это всего лишь обзор основных компонентов HTML. Язык HTML имеет богатый набор тегов и атрибутов, которые позволяют создавать разнообразные веб-страницы с различным содержимым и взаимодействием. Это язык разметки, который является основой для создания веб-страниц и веб-приложений.

### 1.4.3 Язык программирования CSS

CSS (Cascading Style Sheets) [[8]](#_СПИСОК_ЛИТЕРАТУРЫ)- это язык стилей, используемый для определения внешнего вида и форматирования веб-страниц. Он позволяет разработчикам управлять представлением элементов HTML, определяя их цвет, размеры, шрифты, расположение и другие визуальные атрибуты. Вот подробное описание основных аспектов языка CSS:

1. Синтаксис CSS:

- CSS использует селекторы для выбора элементов HTML, к которым будет применяться стиль.

- Стили в CSS определяются через свойства и значения.

- Каждое правило CSS состоит из селектора, за которым следует блок с объявлениями стилей.

2. Селекторы:

- Тип селектора: выбирает все элементы определенного типа. Например, селектор `p` выбирает все абзацы на странице.

- Классовый селектор: выбирает все элементы с определенным классом. Например, селектор `.highlight` выбирает все элементы с классом "highlight".

- Идентификаторный селектор: выбирает элемент с определенным идентификатором. Например, селектор `#header` выбирает элемент с идентификатором "header".

- Комбинированные селекторы: позволяют комбинировать различные условия для выбора элементов. Например, селектор `ul li` выбирает все элементы `<li>`, которые находятся внутри `<ul>`.

3. Свойства и значения:

- CSS предоставляет широкий набор свойств, которые позволяют управлять внешним видом элементов.

- Свойства определяют различные атрибуты элементов, такие как цвет, размеры, шрифты, отступы, границы и фон.

- Каждое свойство имеет соответствующее значение, которое определяет, каким образом свойство будет применяться к элементу.

CSS является важным инструментом для создания привлекательных и функциональных веб-страниц. Он позволяет разработчикам полностью контролировать внешний вид элементов, создавать современные дизайны и обеспечивать адаптивность для различных устройств.

### 1.4.4 Язык программирования React

Функциональные требования [[11]](#_СПИСОК_ЛИТЕРАТУРЫ)к CRM-системе для интернет-магазина этно-бренда “Aisanat”, с учетом использования Next.js для разработки, направлены на обеспечение эффективного управления отношениями с клиентами и повышения уровня удовлетворенности клиентов. Вот переформулированное описание функциональных требований:

1.Управление контактами и лидами:

Система должна предоставлять возможность создания, редактирования и удаления контактных данных клиентов.

Должна быть реализована система управления лидами, позволяющая отслеживать их жизненный цикл от первого контакта до продажи.

Необходимо обеспечить возможность классификации и категоризации клиентов для удобного поиска и организации.

2.Управление продажами и обслуживанием клиентов:

Система должна поддерживать процесс отслеживания продаж, включая создание и управление заказами, отслеживание статуса выполнения заказов и управление возвратами.

Должна быть реализована система обслуживания клиентов, позволяющая обрабатывать запросы и жалобы клиентов, а также предоставлять поддержку через различные каналы связи.

3.Маркетинг и аналитика:

Система должна предоставлять инструменты для проведения маркетинговых кампаний, включая управление кампаниями, отслеживание результатов и анализ поведения клиентов.

Необходимо обеспечить возможность анализа данных о продажах, клиентском поведении и взаимодействии с клиентами для выявления трендов и оптимизации маркетинговых стратегий.

4.Безопасность и конфиденциальность данных:

Система должна обеспечивать безопасность и конфиденциальность данных клиентов, включая аутентификацию пользователей, контроль доступа и шифрование данных.

Должны быть применены меры защиты от несанкционированного доступа и утечки данных.

5.Интеграция с другими системами:

CRM-система должна иметь возможность интеграции с другими информационными системами предприятия, такими как системы управления контентом (CMS), системы электронной коммерции (e-commerce) и системы управления проектами (PM).

Необходимо обеспечить возможность обмена данными между системами для автоматического обновления информации и предотвращения дублирования данных.

6.Отчетность и аналитика:

Система должна предоставлять возможность генерации различных отчетов и аналитических данных по продажам, клиентскому поведению и взаимодействию с клиентами, таких как статистика по количеству продаж, срокам исполнения и другим параметрам.

Пользователи должны иметь доступ к настраиваемым панелям управления и инструментам для анализа данных.

7.Интерфейс пользователя:

Интерфейс CRM-системы должен быть интуитивно понятным и привлекательным, обеспечивая удобство использования, легкость навигации и доступ к основным функциям системы.

Важно обеспечить адаптивность системы для работы на различных устройствах и платформах, включая мобильные устройства и планшеты.

8.Масштабируемость и производительность:

CRM-система должна быть способна обрабатывать большой объем данных и поддерживать одновременную работу нескольких пользователей.

Необходимо обеспечить высокую производительность системы и масштабируемость для возможности расширения функциональности и обработки растущей нагрузки.

9.Архивирование и сохранность данных:

Система должна обеспечивать архивирование данных и сохранность информации в соответствии с требованиями законодательства и внутренними политиками предприятия.

Важно предусмотреть резервное копирование данных и механизмы восстановления системы в случае сбоев или потери информации.

10.Поддержка и обучение:

После внедрения CRM-системы необходимо предоставить обучение для сотрудников, чтобы они могли эффективно использовать систему и получать максимальную пользу от нее.

Необходима поддержка пользователей и техническая поддержка для решения возникающих проблем и вопросов.

Использование Next.js для разработки CRM-системы позволяет создать высокопроизводительное и адаптивное веб-приложение с серверным рендерингом и статической генерацией страниц, что обеспечивает лучшую производительность и SEO. Next.js также поддерживает API-маршруты, что упрощает создание API для интеграции с другими системами и обеспечивает легкую разработку и поддержку/

### 1.4.5 Язык структурированных запросов SQД

Язык программирования SQL[[14]](#_СПИСОК_ЛИТЕРАТУРЫ) (Structured Query Language) является стандартизированным языком запросов, разработанным для работы с реляционными базами данных. Он обеспечивает возможность создания, изменения и управления данными в базах данных.

SQL состоит из различных компонентов и конструкций, позволяющих выполнять различные операции с данными. Вот основные компоненты SQL:

1. Команды DDL (Data Definition Language):

- `CREATE DATABASE`: Создает новую базу данных.

- `CREATE TABLE`: Создает новую таблицу в базе данных.

- `ALTER TABLE`: Изменяет структуру существующей таблицы.

- `DROP TABLE`: Удаляет таблицу из базы данных.

2. Команды DML (Data Manipulation Language):

- `INSERT INTO`: Вставляет новые данные в таблицу.

- `UPDATE`: Обновляет существующие данные в таблице.

- `DELETE FROM`: Удаляет данные из таблицы.

- `SELECT`: Извлекает данные из таблицы.

3. Команды DQL (Data Query Language):

- `SELECT`: Извлекает данные из одной или нескольких таблиц.

- `FROM`: Определяет, из какой таблицы извлекать данные.

- `WHERE`: Устанавливает условия для фильтрации данных.

- `GROUP BY`: Группирует данные по определенным столбцам.

- `HAVING`: Фильтрует данные, основываясь на агрегатных функциях.

- `ORDER BY`: Сортирует данные по указанным столбцам.

- `JOIN`: Объединяет данные из нескольких таблиц.

4. Команды DCL (Data Control Language):

- `GRANT`: Предоставляет определенным пользователям или ролям права доступа к базе данных или таблице.

- `REVOKE`: Отзывает права доступа у пользователей или ролей.

- `COMMIT`: Фиксирует изменения, внесенные в базу данных.

- `ROLLBACK`: Откатывает изменения, внесенные в базу данных.

- `SAVEPOINT`: Устанавливает точку сохранения для возможности отката изменений к этой точке.

5. Команды TCL (Transaction Control Language):

- `BEGIN TRANSACTION`: Начинает новую транзакцию.

- `COMMIT`: Фиксирует изменения, внесенные в базу данных в рамках текущей транзакции.

- `ROLLBACK`: Откатывает изменения, внесенные в базу данных в рамках текущей транзакции.

SQL также поддерживает использование различных операторов и функций для обработки данных:

- Операторы: `AND`, `OR`, `NOT`, `LIKE`, `IN`, `BETWEEN`, `IS NULL`, `UNION`, `INTERSECT`, `EXCEPT` и другие.

- Функции: `COUNT`, `SUM`, `AVG`, `MAX`, `MIN`, `UPPER`, `LOWER`, `CONCAT`, `SUBSTRING`, `DATE`, `TIME` и другие.

SQL позволяет создавать индексы для оптимизации производительности запросов, а также использовать представления (views), хранимые процедуры (stored procedures) и триггеры (triggers) для упрощения и автоматизации операций с базой данных.

Важно отметить, что SQL является декларативным языком, что означает, что вы описываете результат, который хотите получить, а не указываете конкретные шаги для его достижения. SQL-запросы могут быть сложными и мощными, позволяя работать с большим объемом данных и выполнять сложные операции.

SQL широко применяется во множестве областей, включая веб-разработку, анализ данных, бизнес-интеллект и администрирование баз данных. Он является одним из самых популярных языков программирования в области работы с базами данных, и знание SQL является важным навыком для разработчиков, аналитиков и администраторов баз данных.

### 1.4.6 Система для управления базами данных PostgreSQL

Язык программирования PostgreSQL, часто называемый PL/pgSQL (Procedural Language/PostgreSQL), представляет собой процедурный язык, разработанный специально для работы с базой данных PostgreSQL. PL/pgSQL предоставляет расширенные возможности для создания хранимых процедур, функций и триггеров, позволяя разработчикам эффективно управлять данными и выполнением запросов на сервере PostgreSQL.

Основная цель PL/pgSQL - предоставить разработчикам удобный и мощный инструментарий для написания процедурных элементов базы данных. PL/pgSQL имеет синтаксис, схожий с языком программирования SQL, что облегчает его освоение разработчиками, знакомыми с SQL.

Процедуры, функции и триггеры, написанные на PL/pgSQL, компилируются и выполняются на сервере PostgreSQL, что обеспечивает высокую производительность и эффективность при работе с данными. Компиляция PL/pgSQL происходит во время создания объекта базы данных и сохраняет скомпилированный код для последующего использования.

Еще одним важным аспектом PL/pgSQL является его интеграция с базовым языком SQL. Разработчики могут включать SQL-код непосредственно в свои процедуры и функции, что облегчает выполнение сложных запросов и манипуляцию данными внутри процедурных элементов.

Благодаря широким возможностям PL/pgSQL разработчики могут создавать сложные бизнес-логики на стороне сервера, выполнять вычисления и преобразования данных, обрабатывать исключения и выполнять множество других операций непосредственно на базе данных PostgreSQL.

В целом, PL/pgSQL является мощным языком программирования, который позволяет разработчикам полностью использовать возможности базы данных PostgreSQL. Он обеспечивает высокую производительность, гибкость и удобство в написании процедурных элементов, делая PostgreSQL одной из наиболее распространенных и популярных систем управления базами данных.

## 1.5 Обоснование среды разработки

WebStorm по критериям технического, программного и технологического обеспечения

### 1.5.1 Техническое обеспечение

Техническое обеспечение WebStorm включает требования к аппаратному обеспечению и операционной системе, на которой может работать это IDE (Integrated Development Environment). Основные аспекты технического обеспечения:

- Процессор: Минимум 2 ГГц (рекомендуется многоядерный процессор для лучшей производительности).

- Оперативная память: Минимум 4 ГБ (рекомендуется 8 ГБ и более для работы с большими проектами).

- Место на диске: Минимум 2,5 ГБ свободного места (включая место для кэша и временных файлов).

- Дисплей: Разрешение минимум 1024x768 (рекомендуется Full HD для комфортной работы).

- Операционные системы: Поддерживаются Windows 7/8/10/11 (64-бит), macOS 10.13 и новее, Linux (современные дистрибутивы с поддержкой Gnome или KDE).

### 1.5.2. Программное обеспечение

WebStorm предоставляет мощный набор инструментов и функциональных возможностей для разработки, ориентированных на веб-разработку. Ключевые аспекты программного обеспечения:

- Редактор кода: Поддержка HTML, CSS, JavaScript, TypeScript и других веб-технологий. Включает подсветку синтаксиса, автодополнение кода, и функции рефакторинга.

- Интеграция с VCS: Поддержка Git, Mercurial, SVN и других систем контроля версий. Включает удобный интерфейс для управления изменениями и слияниями.

- Отладка: Мощные инструменты отладки для JavaScript, Node.js и других платформ, включая точечные останова, пошаговое выполнение кода, и инспекцию переменных.

- Тестирование: Интеграция с популярными тестовыми фреймворками, такими как Jest, Mocha, Karma, и Jasmine.

- Пакетные менеджеры: Поддержка npm, Yarn, и других пакетных менеджеров для удобного управления зависимостями проектов.

- Инструменты для разработки: Встроенная поддержка для Docker, Webpack, и других современных инструментов и технологий.

### 1.5.3. Технологическое обеспечение

Технологическое обеспечение WebStorm связано с его возможностями для интеграции и поддержки современных технологий и стандартов веб-разработки:

- Поддержка современных фреймворков: WebStorm поддерживает Angular, React, Vue.js и другие популярные фреймворки, предоставляя специальные инструменты и плагины для повышения продуктивности.

- Кросс-платформенность: IDE работает на всех основных операционных системах (Windows, macOS, Linux), что обеспечивает удобство использования для разработчиков с разными предпочтениями.

-Расширяемость: WebStorm поддерживает плагины и позволяет разработчикам создавать свои собственные расширения для улучшения функциональности IDE.

- Облачные возможности: Поддержка облачных сервисов и инструментов для разработки, включая интеграцию с CI/CD пайплайнами и возможность работы с удаленными репозиториями.

- Обновления и поддержка: Регулярные обновления от компании JetBrains обеспечивают актуальность IDE, включая поддержку новых стандартов, языков и фреймворков.

Таким образом, WebStorm предоставляет все необходимое для комфортной и эффективной разработки веб-приложений, сочетая современные технологии с мощными инструментами и широкими возможностями настройки.

# Глава 2 Проектная часть

## 2.1 Концептуальная модель информационных потоков данных

Создание информационной системы для строительных компаний для приема заявок на аренду строительной техники является важным проектом, направленным на оптимизацию управления процессами заказа и улучшение взаимодействия с клиентами. Основные компоненты концептуальной модели информационных потоков включают следующее:

1. Сбор информации о заявках: Система получает запросы и данные от клиентов, содержащие информацию о требуемой технике, сроках аренды и других деталях.

2. Регистрация и обработка заявок: Процесс включает в себя регистрацию новых клиентов в системе, а также обработку их заявок на основе предварительно собранной информации.

3. Маршрутизация и управление заявками: Система определяет и направляет заявки на обработку соответствующим сотрудникам или отделам, учитывая их компетенцию и текущую загруженность.

4. Обработка заявок: Сотрудники компании приступают к обработке заявок, подтверждая доступность техники, уточняя детали заказа и оформляя арендные договоры.

5. Обновление статуса заявок: В процессе обработки заявок сотрудники систематически обновляют информацию о текущем статусе заказов и выполненных действиях.

6. Уведомления и взаимодействие: Система информирует клиентов о статусе их заявок через уведомления, а также обеспечивает возможность взаимодействия сотрудников и клиентов посредством системы.

7. Анализ и отчетность: Система собирает данные для анализа и создания отчетов о процессе обработки заявок, предоставляя руководству информацию о работе компании по аренде строительной техники.

Выбор SQL и Node.js для разработки информационной системы обеспечивает надежное хранение данных и эффективную серверную логику, создавая прочную основу для функционирования системы. Перед приступлением к программной реализации необходимо создать модель приложения, которая может быть представлена в виде диаграмм на языке UML.

## 2.2 Функциональная модель информационных потоков данных

При создании информационной системы для строительных компаний для приема заявок на аренду строительной техники, диаграмма вариантов использования становится ключевым средством для понимания функциональности системы. Этот графический образ иллюстрирует взаимодействие между пользователями и системой в рамках различных сценариев использования. На диаграмме отображаются различные действия, которые могут совершать пользователи, и как эти действия взаимодействуют с самой системой.

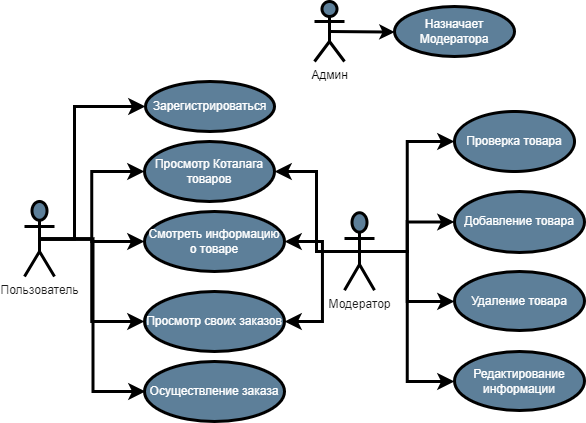
Применительно к созданию информационной системы для строительных компаний, диаграмма вариантов использования позволит наглядно представить, как пользователи будут взаимодействовать с системой при различных сценариях, таких как подача заявок на аренду техники, обработка запросов, управление инвентаризацией и прочее. Это важный инструмент не только для визуализации функциональности системы, но и для последующего анализа и проектирования информационной системы для строительных компаний. 

Рисунок 1 — Диаграмма вариантов использования ИС. [[19]](#_СПИСОК_ЛИТЕРАТУРЫ)

## 2.3 Логическая схема взаимодействия отдельных частей БД

Диаграмма состояний представляет графическое изображение функционирования информационной системы для строительных компаний, предназначенной для приема заявок на аренду строительной техники. Она наглядно демонстрирует различные состояния, в которых может находиться система, а также переходы между этими состояниями в ответ на различные события или условия.

Эта диаграмма позволяет визуализировать и анализировать, как система реагирует на различные события и условия, а также понять поток обработки заявок и изменение состояний системы со временем. Такой инструмент полезен для проектирования и оптимизации информационной системы, обеспечивая эффективное управление заявками и повышение качества обслуживания клиентов в сфере строительства.

Диаграмма состояний также упрощает процесс анализа и оптимизации работы информационной системы, помогая выявить возможные узкие места и улучшить процессы обработки заявок. Она не только предоставляет понимание текущего состояния системы, но и способствует прогнозированию её поведения в ответ на различные сценарии использования, что делает этот инструмент необходимым как на этапе разработки, так и в процессе эксплуатации системы.

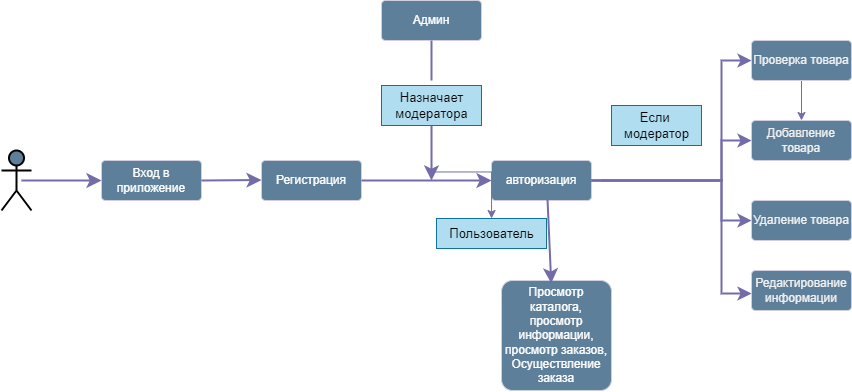


Рисунок 2 — Диаграмма состояний. [[18]](#_СПИСОК_ЛИТЕРАТУРЫ)

## 2.4 Физическая схема взаимодействия отдельных частей БД

ER-диаграмма является ключевым инструментом при создании информационной системы для строительных компаний, предназначенной для приема заявок на аренду строительной техники. Этот инструмент используется для визуализации структуры базы данных, идентификации основных сущностей, их атрибутов и взаимосвязей.

На диаграмме будут представлены следующие элементы:

1. Заявки на аренду: Основные объекты системы, содержащие информацию о заявках на аренду строительной техники.

2. Клиенты: Лица, подающие заявки на аренду и взаимодействующие с системой.

3. Техника: Классификация строительной техники, доступной для аренды.

4. Платежи: Информация о совершенных оплатах за аренду.

5. Доставка: Сведения о процессе доставки арендованной техники клиентам.

6. Другие элементы: Возможно, другие сущности, такие как категории техники, сотрудники, статусы заявок и т. д.

Взаимосвязи между сущностями определяются на основе процессов и логики работы системы. Например, заявка на аренду может быть связана с определенным клиентом, конкретным видом техники и статусом заявки. Также могут быть дополнительные атрибуты, описывающие связи между сущностями.

ER-диаграмма позволяет наглядно представить структуру данных в системе, что помогает разработчикам и аналитикам лучше понять взаимосвязи между элементами информационной системы для строительных компаний. Это необходимый инструмент для проектирования и оптимизации базы данных, что важно для эффективного функционирования системы.

ER-диаграмма также позволяет учитывать специфические потребности строительных компаний, такие как учет технических характеристик арендуемой техники, информация о техническом обслуживании и ремонтах, а также данные о доступности техники на определенные даты.

Кроме того, на диаграмме могут быть отображены дополнительные элементы, такие как отчетность о выручке от аренды, учет затрат на обслуживание техники, а также информация о контрактах и соглашениях с клиентами.

Все эти аспекты помогают создать полнофункциональную информационную систему, способную эффективно управлять процессом аренды строительной техники и обеспечивать удовлетворение потребностей клиентов.

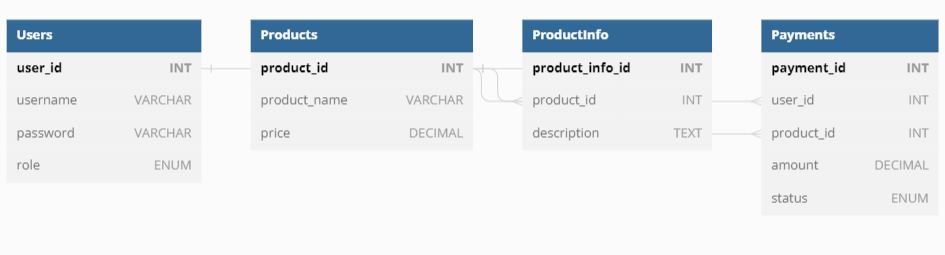


Рисунок 3 — Диаграмма сущностей.

## 2.5 Функциональная модель

На основе данных выше была составлена контекстная диаграмма, показанная на рисунке 4.

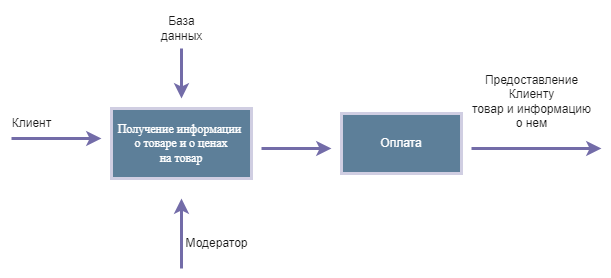


Рисунок 4 — Контекстная диаграмма IDEF0. [[19]](#_СПИСОК_ЛИТЕРАТУРЫ)

## 2.6 Схема технологического процесса

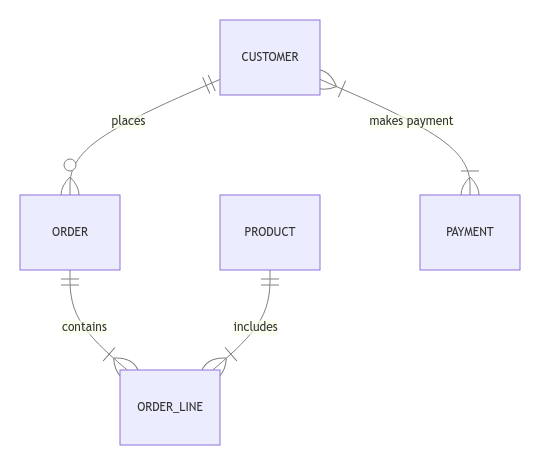


Рисунок 5 — Контекстная диаграмма IDEF1.

На схеме IDEF1 представлена информационная система для строительных компаний, предназначенная для приема заявок на аренду строительной техники. Эта схема описывает следующие элементы и их взаимосвязи:

CLIENT (Клиент): Представляет компанию или частное лицо, которое подает заявку на аренду строительной техники.

REQUEST (Заявка): Представляет собой запрос на аренду строительной техники, поданный клиентом. Заявка содержит информацию о необходимом оборудовании и сроках аренды.

EQUIPMENT (Оборудование): Отражает доступное для аренды строительное оборудование. Каждый элемент оборудования связан с определенной заявкой и относится к определенной категории.

CATEGORY (Категория): Представляет различные категории строительной техники, включая, например, подъемное, землеройное и т. д.

SUBCATEGORY (Подкатегория): Отображает подкатегории строительной техники, входящие в определенную категорию, такие как, например, гусеничные экскаваторы, колесные погрузчики и т. д.

Таким образом, данная схема демонстрирует взаимосвязи между клиентами, заявками, строительной техникой, категориями и подкатегориями в информационной системе для приема заявок на аренду строительной техники.

## 2.7 Схема взаимосвязи программных модулей

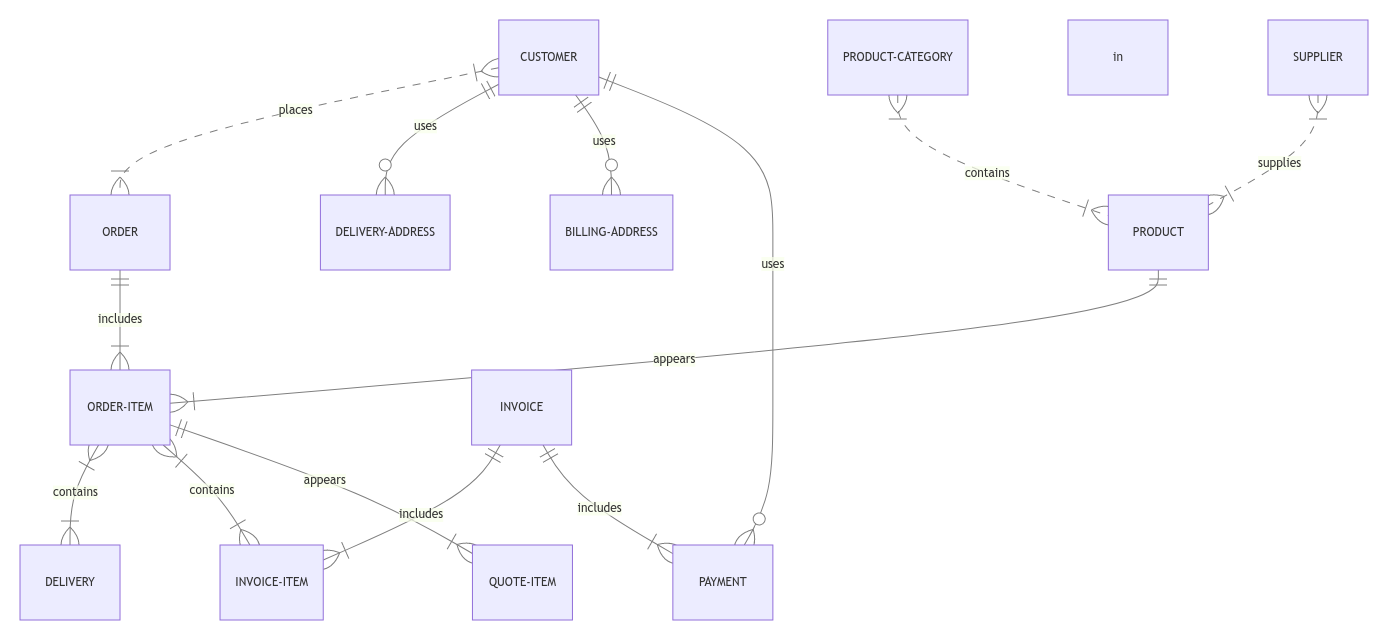


Рисунок 6 — Контекстная диаграмма IDEF2.

Этот чертеж IDEF2 описывает информационную систему приема заказов для строительных компаний, предназначенную для аренды строительной техники. В нем изображены следующие сущности и их взаимосвязи:

- `Клиент`: Устанавливает связь "один-ко-многим" с `Заявкой на аренду` — один клиент может подавать несколько заявок, но каждая заявка связана только с одним клиентом.

- `Заявка на аренду`: Связана с `Оборудованием`, `Оплатой`, `Договором` и `Отделом технической поддержки`. Включает в себя запрос на одно или несколько оборудований, оплату, информацию о договоре и назначение выполнения отделу технической поддержки.

- У сущностей есть атрибуты, описывающие их характеристики, такие как идентификаторы, названия, даты и т. д.

Этот чертеж представляет структуру данных и связи между различными сущностями в информационной системе для строительных компаний. Он может быть использован для проектирования и разработки такой системы, а также для понимания ее основных компонентов и связей между ними.

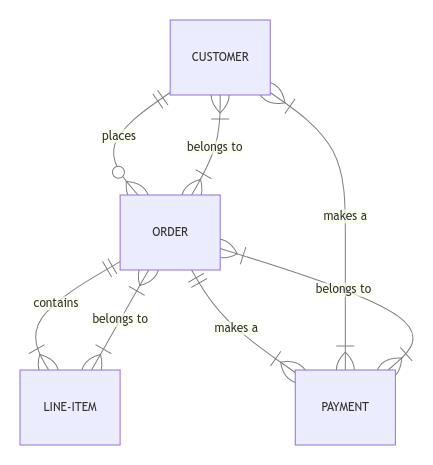


Рисунок 7 — Контекстная диаграмма IDEF3.

Диаграмма контекста является высокоуровневым средством, отображающим информационную систему в контексте ее взаимодействия с внешними сторонами. Она помогает визуализировать взаимосвязи системы с внешними актерами, другими системами или окружающей средой.

При создании информационной системы для строительных компаний, предназначенной для приема заявок на аренду строительной техники, контекстная диаграмма будет отображать процесс приема заказов и взаимодействие с внешними актерами или системами. Этот инструмент поможет понять роль системы в более широком контексте и выделить основные внешние сущности, с которыми она взаимодействует.

## 2.6 Подробное описание проектирования и разработки приложения

Проектирование и разработка информационной системы для строительных компаний для приема заявок на аренду строительной техники является ключевым этапом в оптимизации управления процессами приема заказов и обслуживания клиентов. Рассмотрим этот процесс более подробно.

1. Анализ потребностей

Первым этапом является анализ требований заказчиков и бизнес-процессов строительной компании. Это включает:

- Выявление текущих проблем при приеме заявок на аренду техники.

- Определение ожиданий и требований персонала компании к новой информационной системе.

- Идентификацию необходимых функциональностей системы.

- Планирование интеграции с уже существующими информационными системами компании (если таковые имеются).

2. Проектирование архитектуры

На этом этапе определяется структура и компоненты будущей информационной системы. Важными элементами являются:

- Интерфейс пользователя для приема и управления заявками на аренду.

- Механизм аутентификации и авторизации пользователей.

- Система управления данными о заказах и клиентах.

- Модуль уведомлений для оперативного информирования о статусах заявок.

- База данных для хранения информации о клиентах и заказах.

- Возможность интеграции с другими системами через API.

3. Разработка

На этом этапе происходит создание и реализация информационной системы, включая:

- Создание удобного пользовательского интерфейса.

- Настройку базы данных для хранения информации о заявках и клиентах.

- Разработку бизнес-логики для обработки заявок и управления данными.

- Создание модуля уведомлений.

- Тестирование каждого компонента системы и исправление ошибок.

4. Внедрение и обучение

После завершения разработки системы необходимо приступить к ее внедрению в рабочую среду строительной компании. Этот процесс включает:

- Установку и настройку необходимого оборудования.

- Обучение персонала компании работе с новой системой.

5. Поддержка и сопровождение

Важным этапом является последующая поддержка и развитие информационной системы, включая:

- Поддержку пользователей и решение возникающих проблем.

- Внесение изменений и улучшений в систему на основе обратной связи.

- Регулярное обновление системы для обеспечения безопасности и функциональности.

Это общий обзор процесса проектирования и разработки информационной системы для приема заявок на аренду строительной техники. Каждый этап требует тщательного планирования и выполнения, а также сотрудничества с заинтересованными сторонами для успешного внедрения и использования системы.

# Глава 3 Разработка документации

## 3.1 Руководство пользователю

Пользовательский путь для информационной системы строительной компании для приема заявок на аренду строительной техники:

1. Авторизация и аутентификация:

Пользователь входит в систему с помощью своих учетных данных (логина и пароля), одноразовых паролей или методов биометрической аутентификации.

После успешной аутентификации пользователь получает доступ к персонализированной главной странице.

2. Главная страница:

На главной странице отображается общая информация о системе и персональные уведомления.

Здесь также представлен список задач и проектов, требующих внимания пользователя.

3. Управление контактами и клиентским взаимодействием:

Пользователь может просматривать и редактировать информацию о клиентах, управлять взаимодействием с ними через различные каналы связи (почта, телефон, социальные сети).

Возможности включают создание новых контактов, добавление заметок, назначение задач и отслеживание статуса взаимодействия.

4. Уведомления и напоминания:

Пользователь получает уведомления о новых заявках на аренду строительной техники, задачах, требующих выполнения, и других важных событиях.

Управление уведомлениями позволяет настраивать предпочтения по типу и частоте уведомлений.

5. Поиск и фильтрация:

Система предоставляет функционал поиска и фильтрации заявок на аренду и задач по различным параметрам, таким как статус, дата создания, категория и т. д.

6. Аналитика и отчетность:

Пользователь может получать отчеты и аналитические данные о заявках на аренду строительной техники, продажах, эффективности маркетинговых кампаний и других важных показателях.

7. Настройки и профиль пользователя:

Пользователь может настраивать параметры системы, такие как уведомления, язык интерфейса, а также управлять своим профилем и учетными данными.

8. Поддержка и обратная связь:

В системе предусмотрена возможность обращения за помощью и обратной связи с администраторами системы.

Дополнительные важные шаги в пользовательском пути:

9. Коммуникация и совместная работа:

Пользователи могут обмениваться сообщениями, делиться файлами и документами, а также обсуждать задачи и проекты в режиме реального времени.

10. Интеграция с другими системами:

Возможность интеграции с внешними системами (ERP, бухгалтерия, система управления контентом) для синхронизации данных и автоматизации процессов.

11. Мобильное приложение:

Разработка мобильного приложения для удобства доступа к системе с любого устройства.

12. Обучение и поддержка:

Предоставление обучающих материалов и технической поддержки для помощи пользователям в освоении системы.

13. Безопасность и защита данных:

Обеспечение безопасности данных и конфиденциальности информации пользователей.

14. Аудит и контроль доступа:

Ведение журналов аудита и контроль доступа к данным и функциям системы.

15. Непрерывное улучшение:

Разработка и внедрение новых функций и улучшений на основе обратной связи пользователей и изменений в бизнес-процессах.

Эти аспекты помогут создать эффективную и безопасную информационную систему для строительных компаний, которая будет соответствовать их потребностям и требованиям.

Дополнение к пользовательскому пути для информационной системы строительной компании, учитывая рекомендации и особенности выбора подходящей системы:

1. Определение бизнес-потребностей:

Понимание конкретных потребностей компании в управлении заявками на аренду строительной техники и процессами обработки данных.

2. Исследование рынка программ:

Оценка различных информационных систем на рынке, анализ их функционала, отзывов пользователей, цен и возможностей интеграции с другими инструментами.

3. Учет масштаба бизнеса:

Выбор информационной системы, соответствующей размеру компании и её потребностям, с учетом возможности расширения бизнеса в будущем.

4. Консультация с экспертами:

Получение профессиональной оценки и рекомендаций от IT специалистов или консультантов по выбору информационной системы.

5. Безопасность данных:

Обеспечение высокого уровня защиты конфиденциальной информации компании и данных клиентов.

6. Поддержка клиентов:

Уверенность в том, что выбранная информационная система предоставляет качественную техническую поддержку для оперативного решения проблем и вопросов пользователей.

7. Интеграция с другими инструментами:

Проверка возможности интеграции информационной системы с другими программными продуктами, используемыми в бизнесе, например, с системами управления складом или бухгалтерскими программами.

8. Обучение персонала:

Организация обучения персонала по использованию новой информационной системы для максимальной эффективности её использования.

9. Стоимость реализации и обслуживания:

Расчет всех затрат на внедрение и обслуживание информационной системы, а также стоимость обновлений и технической поддержки.

10. Тестирование перед окончательным выбором:

Проведение тестирования нескольких информационных систем на практике для оценки их удобства использования и соответствия бизнес-требованиям строительной компании.

Эти дополнительные аспекты помогут не только в выборе подходящей информационной системы для строительной компании, но и в успешном внедрении и использовании системы для достижения бизнес-целей.

### 3.1.1 Требования к программному и аппаратному обеспечению

При разработке информационной системы для строительных компаний, предназначенной для приема заявок на аренду строительной техники, следует учитывать следующие требования к программному и аппаратному обеспечению.

1. Программное обеспечение:

Фронтенд-разработка: Использование React и Redux для создания пользовательского интерфейса с поддержкой динамической загрузки данных и управлением состоянием.

База данных: Применение SQL (например, PostgreSQL) для хранения данных о заявках, клиентах и других сущностей, связанных с арендой строительной техники.

Бэкенд-разработка: Разработка серверной части с использованием Node.js или других языков программирования для обработки запросов от фронтенда, взаимодействия с базой данных и выполнения бизнес-логики.

Аутентификация и авторизация: Реализация механизмов аутентификации и авторизации пользователей для обеспечения безопасности и контроля доступа.

2. Аппаратное обеспечение:

Серверное оборудование: Обеспечение серверных ресурсов для обработки запросов и хранения данных, включая выделенные серверы или облачные ресурсы.

Клиентские устройства: Поддержка различных устройств, таких как компьютеры, планшеты и мобильные телефоны, с достаточными вычислительными ресурсами и совместимостью с современными браузерами.

3. Сетевая инфраструктура:

Надежное сетевое соединение: Обеспечение стабильного и высокоскоростного соединения между клиентскими устройствами и серверами для эффективного обмена данными.

Защита данных: Применение механизмов шифрования и протоколов безопасности для защиты информации от несанкционированного доступа.

4. Масштабируемость:

Горизонтальное масштабирование: Обеспечение возможности масштабирования системы для учета роста числа заявок на аренду строительной техники и клиентов.

Высокая доступность: Использование кластеров серверов для обеспечения непрерывной работоспособности системы.

5. Интеграция:

Интеграция с другими системами: Поддержка взаимодействия с другими информационными системами для обеспечения согласованного рабочего процесса.

6. Удобство использования:

Интуитивно понятный интерфейс: Разработка интерфейса, который будет прост в использовании и удовлетворит потребности пользователей строительных компаний при подаче заявок на аренду строительной техники.

Добавление и обновление заявок: Обеспечение возможности добавления, обновления, просмотра статусов, получения уведомлений и формирования отчетов о заявках.

## 3.2 Руководство программисту

### 3.2.1 Проектирование базы данных

Структура информационной системы для строительных компаний для приема заявок на аренду строительной техники:

1. База данных:

- Таблица "Заявки на аренду":

- `id` (INT): уникальный идентификатор заявки (Primary Key).

- `наименование` (VARCHAR): наименование заявки.

- `описание` (TEXT): описание заявки.

- `статус` (VARCHAR): текущий статус заявки (например, "новая", "в обработке", "завершена").

- `ответственный` (INT): идентификатор сотрудника, ответственного за обработку заявки.

- Таблица "Сотрудники":

- `id` (INT): уникальный идентификатор сотрудника (Primary Key).

- `имя` (VARCHAR): имя сотрудника.

- `должность` (VARCHAR): должность сотрудника.

- `email` (VARCHAR): адрес электронной почты сотрудника.

2. Структура папок для проекта на React:

- `src`

- `components`

- `Заявки`

- `СписокЗаявок.js`

- `ДеталиЗаявки.js`

- `ФормаЗаявки.js`

- `Сотрудники`

- `СписокСотрудников.js`

- `ДеталиСотрудника.js`

- `ФормаСотрудника.js`

- `services`

- `api.js`

- `заявки.js`

- `сотрудники.js`

- `App.js`

- `index.js`

Эта структура обеспечивает хранение информации о заявках на аренду и сотрудниках в базе данных. Каждая таблица имеет поля для хранения соответствующих данных. Структура папок для проекта на React разделена по функциональности компонентов. Каждая папка компонента содержит файлы для отображения списка, деталей и формы соответствующей сущности. Папка `services` содержит файлы для взаимодействия с сервером (API) и обработки запросов к заявкам и сотрудникам.

### 3.2.3 Развертывание системы

Для успешного создания информационной системы для строительных компаний, предназначенной для приема заявок на аренду строительной техники, необходимо следовать определенным этапам:

1. Анализ потребностей и планирование:

Провести детальный анализ требований к информационной системе, учитывая специфику строительной отрасли и ожидания клиентов.

Разработать план действий, определяющий этапы, ресурсы и временные рамки проекта.

2. Выбор платформы и функционала:

Определить оптимальную платформу для информационной системы и необходимый функционал с учетом потребностей строительных компаний и особенностей аренды строительной техники.

3. Установка и настройка информационной системы:

Установить и сконфигурировать выбранное программное обеспечение, включая настройку баз данных и пользовательских интерфейсов.

Подготовить интеграцию с веб-платформой и другими системами, если это необходимо.

4. Разработка и тестирование:

Произвести разработку информационной системы, включая настройку базы данных клиентов, систему управления заявками и отчетности.

Провести тестирование системы на соответствие требованиям и исправить выявленные ошибки.

5. Обучение персонала:

Организовать обучение сотрудников строительной компании по использованию новой информационной системы.

Предоставить поддержку и консультации персоналу в процессе внедрения и использования системы.

6. Развертывание и внедрение:

Осуществить развертывание информационной системы в компании, следуя предварительно разработанному плану.

Провести пилотное внедрение и собрать обратную связь от пользователей для дальнейших улучшений.

7. Мониторинг и поддержка:

Настроить систему мониторинга работы информационной системы для оперативного выявления и устранения возможных проблем.

Обеспечить техническую поддержку пользователей и регулярно обновлять систему в соответствии с потребностями и обратной связью пользователей.

## 3.3. Тестирование системы

Тестирование информационной системы для строительных компаний, предназначенной для приема заявок на аренду строительной техники, является ключевым этапом обеспечения её качества и надежности. Этот процесс включает разнообразные виды тестирования, направленные на проверку соответствия системы требованиям и её безошибочную работу.

Основные виды тестирования:

1. Функциональное тестирование:

Проверка функциональности управления заявками: тестирование процессов создания, редактирования, отслеживания и закрытия заявок на аренду техники.

Управление клиентской базой: тестирование функций добавления, редактирования, удаления и поиска информации о клиентах.

Управление взаимодействиями: проверка записи и отслеживания взаимодействий с клиентами (звонки, встречи, электронная почта).

Сегментация клиентов: тестирование возможностей сегментации клиентской базы по различным критериям.

2. Нефункциональное тестирование:

Производительность: проверка времени отклика системы и её стабильности при большом количестве запросов.

Надежность: тестирование устойчивости системы к сбоям и восстановление данных после них.

Безопасность: проведение тестов на выявление уязвимостей и проверка механизмов аутентификации и авторизации пользователей.

3. Тестирование пользовательского интерфейса:

Удобство использования: проверка удобства интерфейса для конечных пользователей.

Кроссбраузерность и кроссплатформенность: тестирование работы системы в различных браузерах и на различных устройствах.

4. Интеграционное тестирование:

Интеграция с внешними системами: проверка взаимодействия с платежными шлюзами, системами доставки и другими внешними сервисами.

5. Регрессионное тестирование:

Проверка системы после внесения изменений, чтобы убедиться в отсутствии возвратных ошибок.

6. Приемочное тестирование:

Проверка системы на соответствие требованиям заказчика и утверждение её им.

Тестирование информационной системы для строительных компаний также играет важную роль в обеспечении эффективности и конкурентоспособности бизнеса. Вот более подробное разъяснение ключевых этапов тестирования:

1. Планирование тестирования: На этом этапе определяется стратегия тестирования и создаются тест-планы и сценарии. Определение объема и приоритетов тестирования позволяет оптимизировать процесс и выделить наиболее важные функции для проверки.

2. Подготовка тестовых данных: Это важный шаг, включающий создание реалистичных данных для различных сценариев использования системы. Тестовые данные должны отражать разнообразные условия работы системы и защищены от утечки конфиденциальной информации.

3. Выполнение тестов: На этом этапе проводятся тесты в соответствии с разработанными сценариями. Результаты фиксируются, а выявленные дефекты передаются на исправление разработчикам.

4. Анализ результатов: После завершения тестирования проводится анализ результатов для выявления проблемных областей и оценки влияния дефектов на работу системы. Это помогает определить приоритеты для исправлений.

5. Отчетность и устранение дефектов: Составляются отчеты о результатах тестирования, включающие описание выявленных проблем и рекомендации по их устранению. Выявленные дефекты передаются команде разработчиков для исправления.

6. Повторное тестирование: После внесения исправлений проводится повторное тестирование для проверки эффективности исправлений и их влияния на работу системы.

Тестирование информационной системы для строительных компаний обеспечивает не только качество и надежность системы, но и повышает доверие пользователей и улучшает их опыт использования. Это позволяет компаниям эффективно управлять заявками на аренду строительной техники, повышая свою конкурентоспособность на рынке.

Процесс тестирования включает в себя планирование, подготовку тестовых данных, выполнение тестов, анализ результатов, отчетность и устранение дефектов. Тестирование информационной системы для строительных компаний обеспечивает высокое качество и надежность системы, что важно для её успешной интеграции в рабочий процесс компаний.

# ГЛАВА 4 Технико-экономическое обоснование разработки программного продукта, который должен содержать следующие подразделы

## 4.1 Определение суммарных затрат на разработку

### 4.1.1 Расчет прямых затрат

К прямым[[2]](#_СПИСОК_ЛИТЕРАТУРЫ) затратам относятся затраты на материалы и комплектующие, необходимые для выполнения данной разработки и полностью включаемые в ее стоимость.

Таблица 1 - Расчет прямых затрат

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Элемент затрат | Цена за шт., сом. | Количество, шт. | Общая стоимость, сом. |
| 1 | Набор канцелярских принадлежностей | 20 | 4 | 80 |
| 2 | Бумага A4 (1 пачка) | 210 | 1 | 210 |
|  | | | Итого: | 290 |

### 4.1.2 Расчет затрат на основную заработную плату

Затраты на основную заработную плату разработчиков определяют как произведение среднедневной заработной платы на трудоемкость разработки по всем работам, выраженную в днях.

Таблица 2 Расчет затрат на основную заработную плату

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Разработчики | Заработная плата, сом./дн. | Шифры выполняемых работ | Общая продолжительность работ, дн. | Основная заработ-ная плата, сом. |
| 1 | Руководитель работы | 150 | 1-2, 4-5, 10-11, 13-14 | 23 | 3450 |
| 2 | Студент | 100 | 1-2, 2-3, 3-4, 5-6, 6-7, 7-8, 8-9, 9-10, 10-11, 11-12, 12-13, 13-14, 14-15 | 55 | 5500 |
|  | | | | Итого: | 8950 |

### 4.1.3 Отчисления на социальные нужды

Из начисленной заработной платы в пользу работников выплачивается 18%, из которых:

15 % направляются в Пенсионный фонд;

2 % в Фонд обязательного медицинского страхования;

1 % в профсоюзный фонд .

Отчисления на социальные нужды (СоцФ) вычисляется по формуле (4.1):

СоцФ = ЗП \* КСоцФ, (4.1)

где ЗП – начисленная заработная плата в пользу работников, сом;

КсоцФ – процент отчислений на социальные нужды от начисленной заработной платы в пользу работников (17,25%);

СоцФ = 8950\*0,1725 = 1543,88 сом.

### 4.1.4 Затраты на содержание и эксплуатацию ПК

Используемые ПК[[3]](#_СПИСОК_ЛИТЕРАТУРЫ) предназначены не только для выполнения данной разработки, поэтому учитываются только текущие затраты на содержание и эксплуатацию.

Расчет стоимости используемых ПК.

Таблица 3 Затраты на содержание и эксплуатацию ПК

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Оборудование | Цена за шт., сом. | Количество, шт. | Общая стоимость, сом. |
| 1 | Системный блок: CPU Intel Core i3-10100 (3.6-4.3GHz), 6MB, LGA1200, Cache L3,EMT64,4 Cores + 8 Threads,Tray,Comet Lake  Клавиатура: Genius KB-320e,  Мышь: A4Tech BT-630,  Монитор: .ACER 19.5  Колонки: Genius SP-S110 2.0  Суммарная мощность: 500 Вт | 39463 | 1 | 39463 |
| 2 | Системный блок: Сpu Intel i3-10100 (3.6-4.3GHz), 6MB, LGA1200, Cache L3,EMT64,4 Cores + 8 Threads,Tray,Comet Lake  Клавиатура: Genius SlimStar 110,  Мышь: Genius NetScroll 120,  Монитор: LG 17”  Колонки: Genius SP-Q06S 2.0  Суммарная мощность: 400 Вт | 19501 | 1 | 19501 |
|  | | | Итого: | 36131 |

В стоимость часа работы ПК включают следующие затраты:

Амортизационные отчисления, приходящиеся на 1 час работы ПК (А), вычисляется по формуле (4.3):

А = Фп \* На / Фв, (4.2)

Где

Фп – первоначальная стоимость ПК, сом;

На – норма амортизации;

Фв – фонд времени работы ПК за год.

Нормы амортизации (На), вычисляются по формуле:

На = [(Фп – Фл) / (Фп \* Tсл)]\*100 , (4.3)

где

Фп – первоначальная стоимость ПК, сом;

Фл – ликвидационная стоимость, сом;

Tсл – срок службы оборудования (10 лет).

Первоначальная стоимость ПК1 (Фп): 39463 сом.

Первоначальная стоимость ПК2 (Фп): 19501 сом.

Ликвидационная стоимость (Фл): 2000 сом.

На= [(39463-2000)/( 39463\*10)]\*100 = 9,5%

На= [(19501-2000)/( 19501\*10)]\*100 = 9,0%

Фонд времени работы ПК (Фв), вычисляются по формуле:

Фв = Чг – (Вг \* Чд + Вм \* 12), (4.4)

где

Чг – число рабочих часов в году (1985 час.);

Вг – время профилактики ежегодное (6 суток);

Чд – число рабочих часов в день (8 час.);

Вм – время профилактики ежемесячное (5 час.);

Фв = 1985 – (6 \* 8 + 5 \* 12) = 1877 (час.)

Расчет амортизационных отчислений (А), по формуле:

А= (39463\*0,095)/1887 = 1,99 сом/ч. (4.5)

А= (19501\*0,09)/1887 = 0,93 сом/ч.

2) Затраты на электрическую энергию, приходящиеся на 1 час работы ПК (Сэл), вычисляется по формуле (5.6):

Сэл = W \* Цэл, (4.6)

где

W – установленная мощность оборудования, кВт/ч (ПК 1=0.5 , ПК 2=0.4);

Цэл - тариф на 1 кВт/ч для прочих непромышленных потребителей:

1,327 сом (без учёта налогов);

1,5 сом (с учетом налогов).

Сэл = 0,5\*1,5 = 0,75 сом/ч.

Сэл = 0,4\*1,5 = 0,6 сом/ч.

3) Затраты на текущий ремонт и профилактику, приходящиеся на 1 час работы ПК (Спр), вычисляется по формуле (4.7):

Cпр = (Фп \* Кзп) / Фв, (4.7)

где

Фп – первоначальная стоимость ПК, сом;

Кзп – процент затрат на текущий ремонт и профилактику от стоимости ПК (5%);

Фв – фонд времени работы ПК за год (1877 час.).

Cпр = 39463\*0,05/1877 = 1,05 сом / ч.

Cпр = 19501\*0,05/1877 = 0,52 сом / ч.

4) Затраты на материалы, необходимые для обеспечения эксплуатации, приходящиеся на 1 час работы ПК (Смэ) , вычисляется по формуле (4.8):

Cмэ = (Фп \* Кзм) / Фв, (4.8)

где

Фп – первоначальная стоимость ПК, сом;

Кзм – процент затрат на материалы, необходимые для обеспечения эксплуатации от стоимости ПК (5%);

Фв – фонд времени работы ПК за год (1877 час.).

Cмэ = 39463\*0,05/1877 = 1,05 сом / ч.

Cмэ = 19501\*0,05/1877 = 0,52 сом / ч.

Затраты в целом, на содержание и эксплуатацию ПК (Спк), вычисляется по формуле (4.9):

Спк = Д \* Ч \* (А + Сэл + Cпр + Cмэ), (4.9)

где

Д - количество дней использования;

Ч - количество часов в день использования;

А - амортизационные отчисления, приходящиеся на 1 час работы, сом;

Сэл - затраты на электрическую энергию, приходящиеся на 1 час работы, сом;

Cпр - затраты на текущий ремонт и профилактику, приходящиеся на 1 час работы, сом;

Cмэ - затраты на материалы, необходимые для обеспечения эксплуатации, на 1 час работы, сом.

Таблица 4 Затраты на 1 час работы оборудований разработчиками

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | Разработчики | | Затраты, приходящиеся на 1 час работы оборудований, сом | | | | Затраты в целом, сом. |
| Дни | Часы | Амортизационные отчисления | Электрическая энергия | Текущий ремонт и профилактика | Материалы для эксплуатации |
| 1 | 23 | 4 | 1,99 | 0,75 | 1,05 | 1,05 | 444,04 |
| 2 | 55 | 8 | 0,93 | 0,6 | 0,52 | 0,52 | 1126,79 |
|  | | | | | | Итого: | 1570,84 |

Спк= 23 \* 4 \* (1,023 + 0,75 + 0,565 + 0,565 ) = 266,99 сом.

Спк= 55 \* 8 \* (0,689 + 0,6 + 0,398 + 0,398 ) = 917,27 сом.

### 4.1.5 Амортизация нематериальных активов

Нематериальные активы [[1]](#_СПИСОК_ЛИТЕРАТУРЫ)- Лицензионное программное обеспечение далее в тексте программное обеспечение (ПО), используемые на ПК предназначены не только для выполнения данной разработки, поэтому учитываются только текущие затраты на амортизацию нематериальных активов (НМА), по формуле (4.10):

НМА = (Фна \* Тисп) / (Тна \* Фв), (4.10):

где

Фна - стоимость лицензии ПО, сом;

Tисп - время использования ПО для выполнения данной разработки, час;

Tна - срок действия лицензии, год (если бессрочная: Tна = 5);

Фв - фонд времени работы ПК (1877 час.).

Таблица 5 Амортизация нематериальных активов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Состав нематериальных активов | Лицензия | | Тисп, час | НМА, сом |
| Цена, сом | Срок, год |
| 1 | Microsoft Office 2007 | 43200 | Бессрочная | 389 | 1790,6 |
|  | | | | Итого: | 1790,6 |

НМА 1 = (43200\*389)/(5\*1877) = 1790,6 сом.

### 4.1.6 Накладные расходы

В их состав входят затраты на ремонт и содержание помещений, расходы на освещение, кондиционирование воздуха, расходы средства связи, расходы на охрану помещения.

Примем размер накладных расходов как 100% от основной заработанной платы. Таким образом, сумма накладных расходов составит 8950 сом.

Суммарные затраты на разработку:

Таблица 6 Накладные расходы

|  |  |  |
| --- | --- | --- |
| № | Элемент затрат | Сумма, сом. |
| 1 | Прямые затраты | 290,00 |
| 2 | Основная заработная плата | 8950,00 |
| 3 | Отчисления на социальные нужды | 1543,88 |
| 4 | Содержание и эксплуатация ПК | 1570,82 |
| 5 | Амортизация нематериальных активов | 1790,6 |
| 6 | Накладные расходы | 8950,00 |
|  | Итого: | 23095,3 |

## 4.2 Расчет ориентировочной цены

Поскольку потенциальная емкость рынка для данной разработки рассчитана, то ориентировочную цену (Цпо), без учета транспортных затрат, рассчитаем по формуле (4.11):

Цпо = C / n + Cдоп + П, (4.11)

где

С - суммарные затраты на разработку, сом;

n - потенциальное число потребителей;

Cдоп - дополнительные затраты, возникающие при продаже разработки каждому потребителю, это упаковка, DVD-диск (50 сом);

П - прибыль, рассчитываемая в процентах от (C /n + Cдоп) (25%).

Цпо = 23095,3/ 50 + 50 + ((23095,3/ 50 + 50)\*0,25) = 689,88 сом.

Вывод: данный проект является эффективным для потребителя и может приносить экономическую выгоду.

# ЗАКЛЮЧЕНИЕ

При разработке информационной системы для строительных компаний, предназначенной для приема заявок на аренду строительной техники, основное внимание было уделено улучшению процессов взаимодействия с клиентами и оптимизации управления бизнес-процессами.

Разработанная система предоставляет современные инструменты для эффективного управления клиентской базой данных, автоматизации процессов приема заявок, отслеживания их статуса и обратной связи с клиентами. Это значительно уменьшает время выполнения оперативных задач и повышает общий уровень обслуживания клиентов.

Кроме того, внедрение системы способствует улучшению аналитики, оптимизации маркетинговых стратегий и повышению лояльности клиентов за счет персонализированного подхода.

Учитывая вышеперечисленное, можно сделать вывод о важности создания и внедрения данной системы для строительных компаний. Ожидается, что это приведет к увеличению объема заказов, улучшению взаимодействия с клиентами и укреплению позиций компании на рынке.

Предложенная система также способствует повышению эффективности управления запасами и планированию производства, что позволяет сократить издержки и оптимизировать процессы аренды строительной техники. Интеграция аналитических инструментов обеспечивает возможность более точного прогнозирования спроса и адаптации бизнес-стратегии в соответствии с рыночными трендами.

Кроме того, создание системы позволяет собирать и анализировать данные о поведении клиентов, что помогает выявить их потребности, а также предоставлять персонализированные рекомендации, увеличивая конверсию и средний чек покупателей.

Общая цель создания информационной системы для строительных компаний состоит в улучшении пользовательского опыта, увеличении прибыли и укреплении позиций компании на рынке аренды строительной техники.

# СПИСОК ЛИТЕРАТУРЫ

1. Советов Б.Я., Цехановский В.В. Информационные технологии: Учебник для вузов. – 3-е изд., стер. – М.: Высшая школа, 2006. – 263 с.6 ил.
2. Рыжиков Ю.И. Вычислительные методы. – СПб.: БХВ-Петербург, 2007. – 400 с.: ил.
3. Беляев А.А., Коротков Э.М. Системология организации: Учеб- ник / Под ред. д-ра экон. наук, проф. Э.М. Короткова. – М.: ИНФРАМ, 2000. – 182 с. С четырьмя и более авторами
4. Eric Newell «Mastering Microsoft Dynamics 365 Implementations»   
   John Wiley & Sons Limited

**Электронный ресурс**

1. "JS для начинающих" - https://learn.javascript.ru/
2. "HTMLBook" - <https://htmlbook.ru/>
3. "CSS гид по Flexbox" - https://css-tricks.ru/css-flexbox-guide/
4. Учебник по HTML и CSS, включающий примеры кода, правила оформления и подробное описание различных элементов и свойств.
5. Интерактивный учебник по JavaScript для начинающих разработчиков.
6. "React документация" - https://ru.reactjs.org/
7. Официальная документация React на русском языке. Включает примеры кода, руководства и рекомендации по разработке на React.
8. "SQL с нуля" - https://sql-ex.ru/
9. Подробное руководство по использованию Flexbox в CSS для создания гибкого и адаптивного макета.
10. "PostgreSQL документация" - https://postgrespro.ru/docs/postgresql
11. "React документация" - https://ru.reactjs.org/
12. "Хороший код" - https://goodcode.io/
13. Блог с статьями, советами и примерами кода по различным аспектам разработки, включая JavaScript, React, SQL и другие технологии.
14. "Хекслет" - <https://ru.hexlet.io/>
15. "Системный анализ и проектирование" [Попов Виктор Юрьевич](https://www.hse.ru/org/persons/24487003), [Сизых Наталья Васильевна](https://www.hse.ru/org/persons/192085683). https://www.hse.ru/edu/courses/292688483
16. "HTML Academy" - https://htmlacademy.ru/
17. Образовательная платформа, предлагающая интерактивные курсы по HTML, CSS и JavaScript, включая курсы по разработке интерфейсов и адаптивного дизайна.
18. The Unified Modeling Language Resource Site (<http://www.uml-diagrams.org/>).
19. Справочник по веб-технологиям, включающий описание основных концепций, терминов и инструкций по использованию различных технологий.
20. "Next.js для начинающих" - https://www.udemy.com/course/nextjs-for-beginners/
21. "Next.js документация" - https://nextjs.org/docs